

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-342135

(43)Date of publication of application : 29. 11. 2002

(51)Int. Cl.

G06F 12/00

(21)Application number : 2001-144622

(71)Applicant : CANON INC

(22)Date of filing : 15. 05. 2001

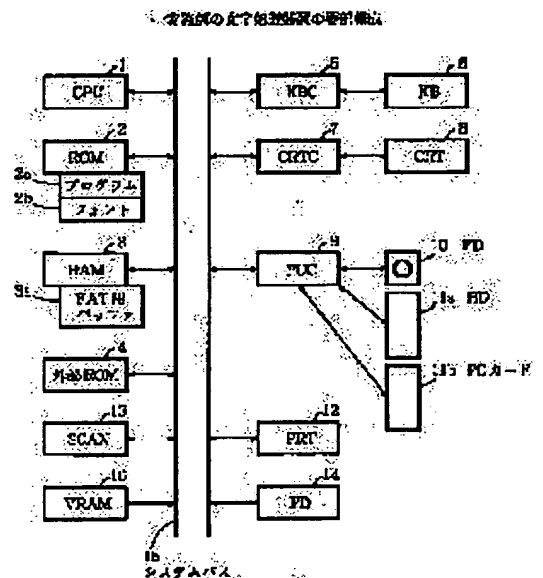
(72)Inventor : SASAHARA NOBUO

(54) DEVICE AND METHOD FOR PROCESSING INFORMATION AND STORAGE MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide an information processor capable of performing file management by suppressing consumption of a memory needed for table information showing the relation between a file and a cluster for performing file management and also minimizing a decrease in a processing speed.

SOLUTION: A CPU (file managing means) 1 decides quantity of FAT information that is read at a time from an FD 10, an HD 11a and a PC card 11b to a memory 3 in accordance with an empty situation of the memory 3. The CPU 1 uses the table information existing in the memory 3 in the case of having to read and write the same FAT information and in the case a FAT information area is the same with a table information area where reading or writing is previously performed. The CPU 1 also writes or reads the FAT information area of the preceding contents to/from the FD 10, the HD 11a and the PC card 11b only when a FAT information area needed this time is different from the preceding FAT information area.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

BEST AVAILABLE COPY

*** NOTICES ***

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] A file management means to perform file management of the storage of the request which has a predetermined file structure, The write-in means which writes in the predetermined table information which shows the relation between the file for performing said file management, and a cluster in said storage, It is the information processor which has the read-out means which reads said table information written in this storage, and is characterized by this file management means determining the amount which reads said table information from said storage to this memory at once with said read-out means according to the empty situation of memory.

[Claim 2] A file management means is an information processor according to claim 1 characterized by identifying an activity or intactness of a cluster, and error information using table information.

[Claim 3] A file management means is an information processor according to claim 1 or 2 characterized by using the table information which exists in memory when it is the table information field same when the need of reading or writing in the same table information comes out as the table information field which read last time or was written in.

[Claim 4] A file management means is an information processor according to claim 3 characterized by carrying out reading appearance from this storage or it makes the table information field of the content write in a storage last time, only when a table information field required this time differs from last time.

[Claim 5] File management of the storage of the request which has a predetermined file structure with a file management means is performed. The predetermined table information which shows the relation between a file for a write-in means to perform said file management to said storage and a cluster is written in. Said table information written in this storage by the read-out means is read. With this file management means The information processing approach characterized by determining the amount which reads said table information from said storage to this memory at once with said read-out means according to the empty situation of memory.

[Claim 6] The information processing approach according to claim 5 characterized by identifying an activity or intactness of a cluster, and error information using table information with a file management means.

[Claim 7] The information processing approach according to claim 5 or 6 characterized by using the table information which exists in memory when it is the same table information field as the table information field which read last time or was written in with the file management means when the need of reading or writing in the same table information had come out.

[Claim 8] The information processing approach according to claim 7 characterized by carrying out reading appearance from this storage or it makes the table information field of the content write in a storage last time with a file management means, only when a table information field required this time differs from last time.

[Claim 9] File management of the storage of the request which has a predetermined file structure with a file management means is performed. The predetermined table information which shows the relation between a file for a write-in means to perform said file management to said storage and a cluster is written in. Said table information written in this storage by the read-out means is read. With this file management means the storage characterized by storing the program for realizing determining said amount which carries out reading appearance, and which is read from said storage to this memory at once with a means for said table information according to the empty situation of memory.

[Claim 10] The storage according to claim 9 characterized by storing the program for realizing identifying an activity or intactness of a cluster, and error information using table information with a file management means.

[Claim 11] The storage according to claim 9 or 10 characterized by storing the program for realizing using the table information which exists in memory when it is the same table information field as the table information field which read last time or was written in with the file management means when the need of reading or writing in the same table information had come out.

[Claim 12] The storage according to claim 11 characterized by storing the program for realizing carrying out reading appearance from this storage or it makes the table information field of the content write in a storage last time with a file management means, only when a table information field required this time differs from last time.

[Translation done.]

NOTICES

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to a storage at the information processor and the information processing approach list which perform file management of desired storages, such as a PC card with the file structure of MSDOS (MSDOS is a trademark), a hard disk, and a floppy (trademark) disk.

[0002]

[Description of the Prior Art] The relation between the file for managing an MSDOS file system conventionally and a cluster is shown, and once the file allocation table (henceforth FAT) for identifying an activity or intactness of a cluster, and error information accumulates the whole of the information in memory, it processes to FAT on memory and has returned FAT after completing required processing.

[0003]

[Problem(s) to be Solved by the Invention] However, it will become, by the time at least memory required for FAT exceeds hundreds of KB with large-capacity-izing of the storage of these days in this case, and the effect to all the memory space of FAT cannot be being disregarded.

[0004] Then, although to make small the amount of memory required for FAT as much as possible, and to manage a file system is desired, making it be only the above, access (R/W) to devices (storage), such as a floppy disk, a hard disk, and a PC card, will increase extremely, and the engine performance, especially processing speed (performance) will only only merely fall.

[0005] This invention aims at providing with a storage the information processor and the information processing approach list which can be made in order to solve the above troubles, can hold down consumption of the memory which the table information which shows the relation between the file for performing file management and a cluster takes, and can suppress lowering of processing speed as much as possible, and can perform file management.

[0006]

[Means for Solving the Problem] A storage is constituted as follows in the information processor and the information processing approach list of this invention.

[0007] (1) A file management means to perform file management of the storage of the request which has a predetermined file structure in an information processor, The write-in means which writes in the predetermined table information which shows the relation between the file for performing said file management, and a cluster in said storage, Having the read-out means which reads said table information written in this storage, this file management means determined the amount which reads said table information from said storage to this memory at once with said read-out means according to the empty situation of memory.

[0008] (2) In the information processor of the above (1), the file management means identified an activity or intactness of a cluster, and error information using table information.

[0009] (3) In the above (1) or the information processor of (2), when it was the table information field same when read the same table information, it writes in or the need of carrying out comes out as the table information field which read last time or was written in, the table information which exists in memory was used for the file management means.

[0010] (4) In the information processor of the above (3), only when a table information field required this time differed from last time, or the file management means made the table information field of the content write in a storage last time, it was made to carry out reading appearance of it from this storage.

[0011] (5) Perform file management of the storage of the request which has a predetermined file structure with a file management means in the information processing approach. The predetermined table information which shows the relation between a file for a write-in means to perform said file management to said storage and a cluster is written in. reading appearance of said table information which carried out reading appearance and was written in this storage by the means is carried out, and this file management means determined said amount which carries out reading appearance and which is read from said storage to this memory at once with a means for said table information according to the empty situation of memory.

[0012] (6) In the information processing approach of the above (5), an activity or intactness of a cluster, and error information were identified using table information the file management means.

[0013] (7) In the above (5) or the information processing approach of (6), when it was the same table information field as the table information field which read last time or was written in with the file management means when read the same table information, it wrote in or the need of carrying out had come out, the table information which exists in memory was used.

[0014] (8) In the information processing approach of the above (7), only when a table information field required this time differed from last time, or it made the table information field of the content write in a storage last time with a file management means, it was

made to carry out reading appearance from this storage.

[0015] (9) Perform file management of the storage of the request which has a predetermined file structure with a file management means. The predetermined table information which shows the relation between a file for a write-in means to perform said file management to said storage and a cluster is written in. Said table information written in this storage by the read-out means is read. With this file management means The program for realizing determining the amount which reads said table information from said storage to this memory at once with said read-out means according to the empty situation of memory was stored in the storage.

[0016] (10) The program for realizing identifying an activity or intactness of a cluster, and error information using table information with a file management means was stored in the storage of the above (9).

[0017] (11) When it was the same table information field as the table information field which read to it last time or was written in it with the file management means when the need of reading or writing in the same table information had appeared in the above (9) or the storage of (10), the program for realizing using the table information which exists in memory was stored.

[0018] (12) Only when a table information field required this time differed in the storage of the above (11) from last time, or it made the storage write the table information field of the content in it last time with a file management means, the program for realizing carrying out reading appearance from this storage was stored.

[0019]

[Embodiment of the Invention] Drawing 1 is the block diagram showing the important section configuration of the character-manipulation equipment (information processor) of one example. In addition, this invention may be carried out as equipment which consists of one device, and can be carried out also as a storage which stored supplying the program for realizing the system which consists of two or more devices, or actuation mentioned later, for example, this program.

[0020] In this drawing, 1 is arithmetic and program control (henceforth CPU), for example, is a microprocessor, it performs the operation for a document or printing processing, controls the various below-mentioned components connected to those buses 15 through the system buses 15, such as an address bus, a control bus, and a data bus, and controls the whole equipment.

[0021] 2 has control-procedure (processing actuation) program data area 2a and font data field 2b which are a read-only memory (henceforth ROM) and are mentioned later, and makes other control programs have memorized. In addition, this ROM2 is made to have memorized the file manager.

[0022] 3 is the random access memory (henceforth RAM) which can write 1-word a 16-bit configuration, and is used for the memory of the various data from various components. In addition, RAM3 has buffer 3a for FAT, and a FAT field is memorized by this RAM3.

[0023] A keyboard controller (KBC) and 6 are keyboards (KB), and the read-only memory (henceforth Exterior ROM) by which 4 was extended outside, and 5 are equipped with the various function keys for directing the various functions to character manipulations, such as for example, alphabetic character symbol keys, such as an alphabet key, a hiragana key, and a katakana key, and conversion, the next candidate. Moving cursor also uses the cursor movement key of KB6 etc.

[0024] 7 is a display-function control unit (CRTC), and controls a display (CRT) 8. The various functions which cannot be displayed only by the keyboard 6 are displayed on this CRT8 as a software menu. Moreover, it is also in displaying the induction message for making an operator perform the function, when the function for which it asks is started ****. The blank paper code showing a line and the cursor for an alphabetic character input are also displayed.

[0025] 9 is a peripheral-device control device (PUC), and controls each device of a floppy disk (FD) 10, (hard disk HD) 11a, and PC card 11b. The program for realizing actuation later mentioned to these FDs10, hard disk 11a, and PC card 11b may be made to store. CPU (file management means)1 manages based on the file manager stored in ROM2 to the file structure of each [these] device.

[0026] 12 is an airline printer (PRT) (henceforth a printer). A printer 12 can choose the printer of the printing method of the request in a connectable printer family, for example, can use an ink jet printer, a laser beam printer, a thermal printer, etc. Furthermore, there is a thing which carried the optical reader (scanner), or a thing which can remove scanner equipment in a printer 12. In the printer equipped with these optical readers, a ruled line round trip printing result can be read.

[0027] 13 is scanner equipment (SCAN). Although this is the same as that of the scanner and functional target which were carried in the printer mentioned above, the places which say that human being operates this scanner and makes the target alphabetic character picture etc. read differ.

[0028] 14 is a pointing device (PD). For example, a mouse etc. is mentioned. Even if migration and the ruled line input of said cursor use this mouse, they are possible. 15 is a system bus for performing various kinds of data and transmission and reception of a control signal.

[0029] The control program of this equipment exists on ROM2 or FD10, HD11a, PC card11b, or RAM3, and processes the object with the control program which CPU1 read through the system bus 15 as mentioned above.

[0030] 16 is VRAM and is the memory only for writing required in order to display text on LCD etc. for a display.

[0031] In the above-mentioned configuration, PUC9 constitutes the read-out means which reads the FAT information written in the write-in means which writes in the FAT information which shows the relation between the file for performing file management to FD10 as a storage, HD11a, and PC card 11b, and a cluster and FD10, HD11a, and PC card 11b.

[0032] The amount in which CPU1 reads FAT information from FD10, HD11a, and PC card11b to memory 3 at once according to the empty situation of memory 3 by the above-mentioned configuration is determined.

[0033] Moreover, when it is the FAT information field same when the need of reading or writing in the same FAT information comes out as the table information field which read last time or was written in, the table information which exists in memory 3 is used for CPU1.

[0034] Moreover, only when a FAT information field required this time differs from last time, or CPU1 writes the FAT information field of the content in FD10, HD11a, and PC card 11b last time, it is read from FD10, HD11a, and PC card11b.

[0035] Next, the actuation of this example mentioned above with reference to drawing 1 and the flow chart of drawing 2 -6 is

explained in detail. The flow chart with which drawing 2 R> 2 shows initialization processing actuation of one example, the flow chart with which drawing 3 shows FAT reading actuation of one example, the flow chart with which drawing 4 shows FAT write-in actuation of one example, the flow chart with which drawing 5 shows reading processing actuation of one example, and drawing 6 are flow charts which show write-in processing actuation of one example.

[0036] How many RAM (only henceforth memory) 3 being used as a FAT field (buffer 3 for FAT a) and processing until it determines are expressed with the flow chart of drawing 2. That is, the empty situation of memory 3 is first investigated at step S101.

[0037] Next, according to the empty situation of memory 3, the magnitude of the memory for FAT (buffer 3 for FAT a) is decided at step S102 (the number of sectors written at once is determined). Although it has "the number of sectors" here, the R/W (access) to the device of FD10, HD11a, and PC card 11b is usually performed in the magnitude of a sector unit. Therefore, it is more efficient to specify the magnitude of access by the integral multiple of a sector. For example, when 1MB or more is vacant, it is decided that a part for 4 sector is made into an access unit etc. If the magnitude of FAT activity memory is determined here, memory will be actually secured at step S103 (memory buffer acquisition for several sector minutes). For example, if it is the device of 512 bytes of 1 sector, and it is the former, the memory for FAT which was required hundreds of KB can be managed with at least 512 bytes.

[0038] The flow chart of drawing 3 shows the flow of the processing which reads FAT. The FAT field where it corresponds there since it is specified what cluster (unit of the magnitude which MSDOS has managed logically, and to write) should be read in front of step S201 with a natural thing will be read.

[0039] First, it reads and writes in, and the reading actuation includes and confirms whether to be that it is the first. If the first, since nothing close is in buffer 3 for FAT a gained at step S103 of drawing 2, it is necessary to say existence but to read FAT. It drops "it is a flag for the first time" on step S202 first. Since a device is specified when usually writing by logical block NO (number which carried out numbering sequentially to the sector) called LBN, it performs conversion to LBN from a cluster first. And it reads (step S204).

[0040] Step S204 (step S401 of drawing 5) is actuation which reads to a device physically. The data read with the natural thing are stored in said buffer 3a for FAT. At step S207, from buffer 3 for FAT a, information over a cluster is considered as the result, and is returned.

[0041] The sector (it is two or more sector reading **** a case the LBN number) now read at step S208 is kept in mind. It compares whether it carries out also at step S201, and in not being the first access, LBN is in the same range at step S205 next. Since it is not necessary to read from a device (FD10, HD11a, PC card 11b) anew if the same, the information in buffer 3a for FAT is used as it is. Therefore, what is necessary is just to go to step S207. When there is no cluster which it is going to read now in the range of LBN of step S206, it is necessary to read from a device (FD10, HD11a, PC card 11b) anew.

[0042] Write-in processing of step S206 (flow chart of drawing 6) performs processing which once writes the data which already exist in buffer 3a for FET in a device (FD10, HD11a, PC card 11b) here. However, if it becomes as it is, it is not necessary to write in the read FAT information. Then, in step S206 (flow chart of drawing 6), it checks [whether it is the same as the time of the data in buffer 3a for FAT reading, and] at step S501 of drawing 6. Writing is needed when different. And after processing the data which suited buffer 3a for FAT, in order to read LBN corresponding to a new cluster, it flies to step S203. The above is FAT reading processing.

[0043] Next, FAT write-in processing is explained. It confirms first whether to be the first writing as well as [FAT write-in processing] reading processing at step S301 of drawing 4. When the first, after reading the sector, dropping "it is a flag for the first time" on step S305 after computing LBN equivalent to the cluster written in at step S303, writing in at step S309 further and setting a flag, the value which should write to buffer 3a for FAT is assigned to step S310.

[0044] When "it is a flag for the first time" does not stand at step S301, LBN confirms whether it is the same as what is already in buffer 3a for FAT at step S302. If the same, it is only passing along step S309 and step S310. When LBN(s) differ, the content which is already in buffer 3a for FAT is written in a device (FD10, HD11a, PC card 11b) (step S306). And new LBN is read and the value after are recording is changed into buffer 3a for FAT.

[0045] The flow chart of drawing 6 shows write-in processing to the device (FD10, HD11a, PC card 11b). It judges whether it writes in at step S501 and the flag stands, and if the write-in flag stands, the sector containing LBN which corresponds at step S502 will be written in. Then, it writes in at step S503 and a flag is cleared. Thus, what (step S501) a write-in flag is checked for is made not to perform useless writing.

[0046] Thus, in this example, the activities of the memory 3 which FAT takes can be reduced substantially, lowering of the processing speed accompanying it can be prevented as much as possible, and file management which was excellent in performance can be performed.

[0047]

[Effect of the Invention] As explained above, according to this invention, it is effective in the ability to hold down consumption of the memory which the table information which shows the relation between the file for performing file management and a cluster takes, and suppress lowering of processing speed as much as possible, and perform file management.

*** NOTICES ***

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] The block diagram showing the important section configuration of the character-manipulation equipment of one example

[Drawing 2] The flow chart which shows initialization processing actuation of one example

[Drawing 3] The flow chart which shows FAT reading actuation of one example

[Drawing 4] The flow chart which shows FAT write-in processing actuation of one example

[Drawing 5] The flow chart which shows reading processing actuation of one example

[Drawing 6] The flow chart which shows write-in processing actuation of one example

[Description of Notations]

1 CPU

2 ROM

3 RAM

3a The buffer for FAT

9 PUC

10 Floppy Disk (FD)

11a Hard disk (HD)

11b PC card

15 System Bus

[Translation done.]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2002-342135
(P2002-342135A)

(43) 公開日 平成14年11月29日(2002.11.29)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード(参考)
G 0 6 F 12/00	5 1 4	G 0 6 F 12/00	5 1 4 R 5 B 0 8 2
	5 0 1		5 0 1 H
	5 2 0		6 2 0 J

審査請求 未請求 請求項の数12 O L (全 6 頁)

(21) 出願番号 特願2001-144622(P2001-144622)

(22) 出願日 平成13年5月15日(2001.5.15)

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 笹原 伸雄

東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

(74) 代理人 100066061

弁理士 丹羽 宏之 (外1名)

Fターム(参考) 5B082 CA01 EA01

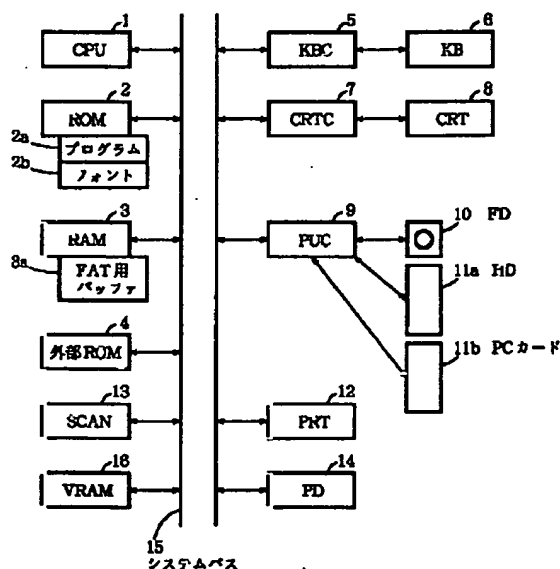
(54) 【発明の名称】 情報処理装置及び情報処理方法並びに記憶媒体

(57) 【要約】

【課題】 ファイル管理を行うためのファイルとクラスタとの関係を示すテーブル情報に要するメモリの消費を抑え、かつ処理速度の低下を極力抑えてファイル管理を行うことができる情報処理装置を提供する。

【解決手段】 CPU(ファイル管理手段)1は、メモリ3の空き状況に応じてFAT情報をFD10、HD11a、PCカード11bからメモリ3に一度に読み出す量を決定する。またCPU1は、同一のFAT情報を読み出したり書き込んだりする必要が出てきた場合、前回読み出したまたは書き込んだテーブル情報領域と同一のFAT情報領域の場合、メモリ3に存在するテーブル情報を使用する。またCPU1は、今回必要なFAT情報領域が前回と異なった場合のみ前回内容のFAT情報領域をFD10、HD11a、PCカード11bに書き込むまたはFD10、HD11a、PCカード11bから読み出す。

一実施例の文字処理装置の要部構成



【特許請求の範囲】

【請求項1】 所定のファイル構造を有する所望の記憶媒体のファイル管理を行うファイル管理手段と、前記記憶媒体に前記ファイル管理を行うためのファイルとクラスタとの関係を示す所定のテーブル情報を書き込む書き込み手段と、該記憶媒体に書き込まれた前記テーブル情報を読み出す読み出し手段とを有し、該ファイル管理手段は、メモリの空き状況に応じて前記テーブル情報を前記読み出し手段により前記記憶媒体から該メモリに一度に読み出す量を決定することを特徴とする情報処理装置。

【請求項2】 ファイル管理手段は、テーブル情報によりクラスタの使用あるいは未使用、エラー情報を識別することを特徴とする請求項1記載の情報処理装置。

【請求項3】 ファイル管理手段は、同一のテーブル情報を読み出したり書き込んだりする必要がある場合、前回読み出したまたは書き込んだテーブル情報領域と同一のテーブル情報領域の場合、メモリに存在するテーブル情報を使用することを特徴とする請求項1または2記載の情報処理装置。

【請求項4】 ファイル管理手段は、今回必要なテーブル情報領域が前回と異なった場合のみ前回内容のテーブル情報領域を記憶媒体に書き込ませるまたは該記憶媒体から読み出させることを特徴とする請求項3記載の情報処理装置。

【請求項5】 ファイル管理手段により所定のファイル構造を有する所望の記憶媒体のファイル管理を行い、書き込み手段により前記記憶媒体に前記ファイル管理を行うためのファイルとクラスタとの関係を示す所定のテーブル情報を書き込み、読み出し手段により該記憶媒体に書き込まれた前記テーブル情報を読み出し、該ファイル管理手段により、メモリの空き状況に応じて前記テーブル情報を前記読み出し手段により前記記憶媒体から該メモリに一度に読み出す量を決定することを特徴とする情報処理方法。

【請求項6】 ファイル管理手段により、テーブル情報によりクラスタの使用あるいは未使用、エラー情報を識別することを特徴とする請求項5記載の情報処理方法。

【請求項7】 ファイル管理手段により、同一のテーブル情報を読み出したり書き込んだりする必要がある場合、前回読み出したまたは書き込んだテーブル情報領域と同一のテーブル情報領域の場合、メモリに存在するテーブル情報を使用することを特徴とする請求項5または6記載の情報処理方法。

【請求項8】 ファイル管理手段により、今回必要なテーブル情報領域が前回と異なった場合のみ前回内容のテーブル情報領域を記憶媒体に書き込ませるまたは該記憶媒体から読み出させることを特徴とする請求項7記載の情報処理方法。

【請求項9】 ファイル管理手段により所定のファイル

構造を有する所望の記憶媒体のファイル管理を行い、書き込み手段により前記記憶媒体に前記ファイル管理を行うためのファイルとクラスタとの関係を示す所定のテーブル情報を書き込み、読み出し手段により該記憶媒体に書き込まれた前記テーブル情報を読み出し、該ファイル管理手段により、メモリの空き状況に応じて前記テーブル情報を前記読み出し手段により前記記憶媒体から該メモリに一度に読み出す量を決定することを特徴とする記憶媒体。

【請求項10】 ファイル管理手段により、テーブル情報によりクラスタの使用あるいは未使用、エラー情報を識別することを特徴とする請求項9記載の記憶媒体。

【請求項11】 ファイル管理手段により、同一のテーブル情報を読み出したり書き込んだりする必要がある場合、前回読み出したまたは書き込んだテーブル情報領域と同一のテーブル情報領域の場合、メモリに存在するテーブル情報を使用することを特徴とする請求項9または10記載の記憶媒体。

【請求項12】 ファイル管理手段により、今回必要なテーブル情報領域が前回と異なった場合のみ前回内容のテーブル情報領域を記憶媒体に書き込ませるまたは該記憶媒体から読み出させることを特徴とする請求項11記載の記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、例えばMSDOS（MSDOSは登録商標）のファイル構造を持つPCカード、ハードディスク、フロッピー（登録商標）ディスク等の所望の記憶媒体のファイル管理を行う情報処理装置及び情報処理方法並びに記憶媒体に関するものである。

【0002】

【従来の技術】従来、MSDOSファイルシステムを管理するためのファイルとクラスタとの関係を示し、クラスタの使用あるいは未使用、エラー情報を識別するためのファイル・アロケーション・テーブル（以下、FATという）は、その情報をすべてメモリに一旦蓄積した後、メモリ上のFATに対して処理を行い、必要な処理が終了後、FATを書き戻すようにしている。

【0003】

【発明が解決しようとする課題】しかしながら、この場合昨今の記憶装置の大容量化に伴い、FATに必要なメモリだけでも数百KBを超えるまでになり、FATの全メモリ容量に対する影響は無視できないものになってきている。

【0004】そこで、FATに必要なメモリ量を極力小さくしてファイルシステムを管理することが望まれる

が、ただ単に前記のようにするだけでは、フロッピーディスク、ハードディスク、PCカードなどのデバイス（記憶媒体）に対するアクセス（読み書き）が極端に増え、性能、特に処理速度（パフォーマンス）が低下してしまう。

【0005】本発明は、上記のような問題点を解決するためになされたものであり、ファイル管理を行うためのファイルとクラスタとの関係を示すテーブル情報に要するメモリの消費を抑え、かつ処理速度の低下を極力抑えてファイル管理を行うことができる情報処理装置及び情報処理方法並びに記憶媒体を提供することを目的とする。

【0006】

【課題を解決するための手段】本発明の情報処理装置及び情報処理方法並びに記憶媒体は、次のように構成したものである。

【0007】（１）情報処理装置において、所定のファイル構造を有する所望の記憶媒体のファイル管理を行うファイル管理手段と、前記記憶媒体に前記ファイル管理を行うためのファイルとクラスタとの関係を示す所定のテーブル情報を書き込む書き込み手段と、該記憶媒体に書き込まれた前記テーブル情報を読み出す読み出し手段とを有し、該ファイル管理手段は、メモリの空き状況に応じて前記テーブル情報を前記読み出し手段により前記記憶媒体から該メモリに一度に読み出す量を決定するようにした。

【0008】（２）上記（１）の情報処理装置において、ファイル管理手段は、テーブル情報によりクラスタの使用あるいは未使用、エラー情報を識別するようにした。

【0009】（３）上記（１）または（２）の情報処理装置において、ファイル管理手段は、同一のテーブル情報を読み出したり書き込んだりする必要が出てきた場合、前回読み出したまたは書き込んだテーブル情報領域と同一のテーブル情報領域の場合、メモリに存在するテーブル情報を使用するようにした。

【0010】（４）上記（３）の情報処理装置において、ファイル管理手段は、今回必要なテーブル情報領域が前回と異なった場合のみ前回内容のテーブル情報領域を記憶媒体に書き込ませるまたは該記憶媒体から読み出させるようにした。

【0011】（５）情報処理方法において、ファイル管理手段により所定のファイル構造を有する所望の記憶媒体のファイル管理を行い、書き込み手段により前記記憶媒体に前記ファイル管理を行うためのファイルとクラスタとの関係を示す所定のテーブル情報を書き込み、読み出し手段により該記憶媒体に書き込まれた前記テーブル情報を読み出し、該ファイル管理手段により、メモリの空き状況に応じて前記テーブル情報を前記読み出し手段により前記記憶媒体から該メモリに一度に読み出す量を

決定するようにした。

【0012】（６）上記（５）の情報処理方法において、ファイル管理手段により、テーブル情報によりクラスタの使用あるいは未使用、エラー情報を識別するようにした。

【0013】（７）上記（５）または（６）の情報処理方法において、ファイル管理手段により、同一のテーブル情報を読み出したり書き込んだりする必要が出てきた場合、前回読み出したまたは書き込んだテーブル情報領域と同一のテーブル情報領域の場合、メモリに存在するテーブル情報を使用するようにした。

【0014】（８）上記（７）の情報処理方法において、ファイル管理手段により、今回必要なテーブル情報領域が前回と異なった場合のみ前回内容のテーブル情報領域を記憶媒体に書き込ませるまたは該記憶媒体から読み出させるようにした。

【0015】（９）ファイル管理手段により所定のファイル構造を有する所望の記憶媒体のファイル管理を行い、書き込み手段により前記記憶媒体に前記ファイル管理を行うためのファイルとクラスタとの関係を示す所定のテーブル情報を書き込み、読み出し手段により該記憶媒体に書き込まれた前記テーブル情報を読み出し、該ファイル管理手段により、メモリの空き状況に応じて前記テーブル情報を前記読み出し手段により前記記憶媒体から該メモリに一度に読み出す量を決定することを実現させるためのプログラムを記憶媒体に格納した。

【0016】（１０）上記（９）の記憶媒体に、ファイル管理手段により、テーブル情報によりクラスタの使用あるいは未使用、エラー情報を識別することを実現させるためのプログラムを格納した。

【0017】（１１）上記（９）または（１０）の記憶媒体に、ファイル管理手段により、同一のテーブル情報を読み出したり書き込んだりする必要が出てきた場合、前回読み出したまたは書き込んだテーブル情報領域と同一のテーブル情報領域の場合、メモリに存在するテーブル情報を使用することを実現させるためのプログラムを格納した。

【0018】（１２）上記（１１）の記憶媒体に、ファイル管理手段により、今回必要なテーブル情報領域が前回と異なった場合のみ前回内容のテーブル情報領域を記憶媒体に書き込ませるまたは該記憶媒体から読み出させることを実現させるためのプログラムを格納した。

【0019】

【発明の実施の形態】図１は一実施例の文字処理装置（情報処理装置）の要部構成を示すブロック図である。なお、本発明は、１つの機器からなる装置として実施してもよいし、複数の機器から構成されるシステムあるいは後述する動作を実現させるためのプログラムを供給すること、例えば該プログラムを格納した記憶媒体としても実施可能である。

【0020】同図において、1は中央演算処理装置（以下、CPUという）で、例えばマイクロプロセッサであり、文書や印刷処理のための演算を行い、アドレスバス、コントロールバス、データバス等のシステムバス15を介してそれらのバス15に接続された後述の各種構成要素を制御して装置全体の制御を行う。

【0021】2は読み出し専用メモリ（以下、ROMという）であり、後述する制御手順（処理動作）プログラムデータ領域2aやフォントデータ領域2bを有し、その他の制御プログラムを記憶させてある。なお、ファイルマネージャはこのROM2に記憶させてある。

【0022】3は例えば1ワード16ビットの構成の読み書き可能なランダムアクセスメモリ（以下、RAMという）であり、各種構成要素からの各種データの一時記憶に用いる。なお、RAM3はFAT用バッファ3aを有し、FAT領域はこのRAM3に記憶される。

【0023】4は外部に増設された読み出し専用メモリ（以下、外部ROMという）、5はキーボード制御装置（KBC）、6はキーボード（KB）であり、アルファベットキー、ひらがなキー、カタカナキー等の文字記号キー、及び例えば変換・次候補などの文字処理に対する各種機能を指示するための各種ファンクションキーを備えている。カーソルを移動させたりするのもKB6のカーソル移動キーなどを使う。

【0024】7は表示機能制御装置（CRTC）であり、表示装置（CRT）8を制御する。このCRT8にはキーボード6だけでは表示しきれない多様な機能をソフトメニューとして表示したりする。また所望する機能を起動させた場合、作業者にその機能を実行させるための誘導メッセージを表示させたりもする。行を表す白紙コードや文字入力のためのカーソルも表示させたりする。

【0025】9は周辺装置制御装置（PUC）であり、フロッピーディスク（FD）10、ハードディスク（HD）11a、PCカード11bの各デバイスを制御する。これらFD10、ハードディスク11a、PCカード11bに後述する動作を実現するためのプログラムを格納させておいてもよい。これら各デバイスのファイル構造に対してROM2に格納されたファイルマネージャに基づいてCPU（ファイル管理手段）1が管理する。

【0026】12は印刷装置（PRT）（以下、プリンタという）である。プリンタ12は接続可能なプリンタファミリー中の所望の印字方式のプリンタを選択でき、例えばインクジェットプリンタ、レーザビームプリンタ、サーマルプリンタ等を使用することができる。さらにプリンタ12には光学読み取り装置（スキャナ）を搭載したもの、あるいはスキャナ装置をとりはずしできるものなどがある。これらの光学読み取り装置を備えたプリンタにおいて野線往復印字結果を読み取ることができる。

【0027】13はスキャナ装置（SCAN）である。

これは前述したプリンタに搭載されたスキャナと機能的には同様のものであるが、人間が本スキャナを操作して対象の文字絵などを読み取らせるところが異なっている。

【0028】14はポインティングデバイス（PD）である。例えばマウスなどが挙げられる。前記カーソルの移動や野線入力はこのマウスを利用してもできる。15は各種のデータ及び制御信号の送受信を行うためのシステムバスである。

【0029】上述した通り本装置の制御プログラムはROM2、またはFD10、HD11a、PCカード11bあるいはRAM3上に存在し、CPU1がシステムバス15を通じて読み出した制御プログラムにより目的の処理を行う。

【0030】16はVRAMであり、文字情報をLCDなどに表示させるために必要な表示用書き込み専用メモリである。

【0031】上記構成において、PUC9は記憶媒体としてのFD10、HD11a、PCカード11bにファイル管理を行うためのファイルとクラスタとの関係を示すFAT情報を書き込む書き込み手段及びFD10、HD11a、PCカード11bに書き込まれたFAT情報を読み出す読み出し手段とを構成している。

【0032】上記構成によりCPU1は、メモリ3の空き状況に応じてFAT情報をFD10、HD11a、PCカード11bからメモリ3に一度に読み出す量を決定する。

【0033】またCPU1は、同一のFAT情報を読み出したり書き込んだりする必要が出てきた場合、前回読み出したまたは書き込んだテーブル情報領域と同一のFAT情報領域の場合、メモリ3に存在するテーブル情報を使用する。

【0034】またCPU1は、今回必要なFAT情報領域が前回と異なった場合のみ前回内容のFAT情報領域をFD10、HD11a、PCカード11bに書き込むまたはFD10、HD11a、PCカード11bから読み出す。

【0035】次に図1及び図2～6のフローチャートを参照して上述した本実施例の動作を詳しく説明する。図2は一実施例の初期化処理動作を示すフローチャート、図3は一実施例のFAT読み込み動作を示すフローチャート、図4は一実施例のFAT書き込み動作を示すフローチャート、図5は一実施例の読み込み処理動作を示すフローチャート、図6は一実施例の書き込み処理動作を示すフローチャートである。

【0036】図2のフローチャートでは、FAT領域（FAT用バッファ3a）としてRAM（以下、単にメモリという）3をどの程度使用するか決定するまでの処理を表している。すなわち、ステップS101でまずメモリ3の空き状況を調べる。

【0037】次にステップS102でメモリ3の空き状況に応じてFAT用メモリ（FAT用バッファ3a）の大きさを決める（1回で読み書きするセクタ数を決定）。ここではセクタ数となっているが、通常、FD10、HD11a、PCカード11bのデバイスに対する読み書き（アクセス）は、セクタ単位で行われる。したがってセクタの整数倍でアクセスの大きさを指定した方が効率がよい。例えば、1MB以上空いている場合は、4セクタ分をアクセス単位とする、などと決めておくのである。ここでFAT使用メモリの大きさが決定されたらステップS103で実際にメモリを確保する（セクタ数分のメモリバッファ取得）。例えば、1セクタ512バイトのデバイスであるならば、従来だったら数百KB必要だったFAT用メモリが最低512バイトで済むようになる。

【0038】図3のフローチャートでは、FATを読み込む処理のフローを示している。当然のことながらステップS201の前で、どこクラスタ（論理的にMSDOSが管理している読み書きする大きさの単位）を読むべきか指定されてくるのでそこに相当するFAT領域を読むことになる。

【0039】まずその読み込み動作が読み込み書き込み含めて初めてであるのかどうかをチェックする。初めてであれば図2のステップS103で獲得したFAT用バッファ3aには何も入っていないから有無を言わずFATを読む必要がある。まずステップS202で初めてフラグを落しておく。デバイスは通常LBNという論理ブロックNO（セクタに対してシーケンシャルにナンバリングした番号）で読み書きする場合に指定するのでまずクラスタからLBNへの変換を行う。そして読み込みを行う（ステップS204）。

【0040】ステップS204（図5のステップS401）は物理的にデバイスに対して読み込みを行う動作である。当然のことながら読み込んだデータは前記FAT用バッファ3aに蓄積される。ステップS207では、FAT用バッファ3aから、クラスタに対する情報をその結果として返却する。

【0041】ステップS208で今読んだセクタ（複数セクタ読み込んだ場合には、そのLBN番号）を覚えておく。ステップS201でもし初めてのアクセスでない場合には次にステップS205でLBNが同じ範囲にあるかどうか比較する。もし同じであればあらかじめデバイス（FD10、HD11a、PCカード11b）から読み込む必要はないので、FAT用バッファ3aにある情報をそのまま使う。したがってステップS207に行けばよい。もし今読もうとしているクラスタがステップS206のLBNの範囲にない場合、あらかじめデバイス（FD10、HD11a、PCカード11b）から読む必要がある。

【0042】ここでステップS206（図6のフロー

チャート）の書き込み処理は、すでにFET用バッファ3aに存在するデータを一旦デバイス（FD10、HD11a、PCカード11b）に書き込む処理を行う。しかしながら読み込んだFAT情報をそのままならば書き込む必要はない。そこでステップS206（図6のフローチャート）において、FAT用バッファ3aにあるデータが読み込んだ時と同じかどうか図6のステップS501でチェックする。もし違う場合は、書き込みが必要となる。そしてFAT用バッファ3aにあったデータの処理をした後、新しいクラスタに対応したLBNを読み込むためステップS203に飛ぶ。以上がFAT読み込み処理である。

【0043】次にFAT書き込み処理を説明する。FAT書き込み処理も読み込み処理と同じように初めての書き込みかどうかをまず図4のステップS301でチェックする。初めての場合、ステップS303で書き込むクラスタに相当するLBNを算出後、そのセクタを読み込みステップS305で初めてフラグを落とし、更にステップS309で書き込みフラグを立てた後、ステップS310でFAT用バッファ3aに書くべき値を代入する。

【0044】ステップS301で初めてフラグが立っていない場合、ステップS302でLBNがすでにFAT用バッファ3aにあるものと同じかどうかチェックする。同じであればステップS309とステップS310を通るのみである。もしLBNが異なる場合は、すでにFAT用バッファ3aにある内容をデバイス（FD10、HD11a、PCカード11b）に書き込む（ステップS306）。そして新しいLBNを読み込んでFAT用バッファ3aに蓄積後値を変更する。

【0045】図6のフローチャートはデバイス（FD10、HD11a、PCカード11b）に書き込み処理を示している。ステップS501で書き込みフラグが立っているかを判断し、書き込みフラグが立っていればステップS502で該当するLBNを含むセクタを書き込む。その後、ステップS503で書き込みフラグをOFFにする。このように書き込みフラグをチェックする（ステップS501）ことによって無駄な書き込みを行わないようにしている。

【0046】このように本実施例では、FATに要するメモリ3の使用を大幅に削減し、それに伴う処理速度の低下を極力防ぎ、パフォーマンスの優れたファイル管理を行うことができる。

【0047】

【発明の効果】以上説明したように、本発明によれば、ファイル管理を行うためのファイルとクラスタとの関係を示すテーブル情報に要するメモリの消費を抑え、かつ処理速度の低下を極力抑えてファイル管理を行うことができるという効果がある。

【図面の簡単な説明】

【図1】 一実施例の文字処理装置の要部構成を示すブロック図

【図2】 一実施例の初期化処理動作を示すフローチャート

【図3】 一実施例のFAT読み込み動作を示すフローチャート

【図4】 一実施例のFAT書き込み処理動作を示すフローチャート

【図5】 一実施例の読み込み処理動作を示すフローチャート

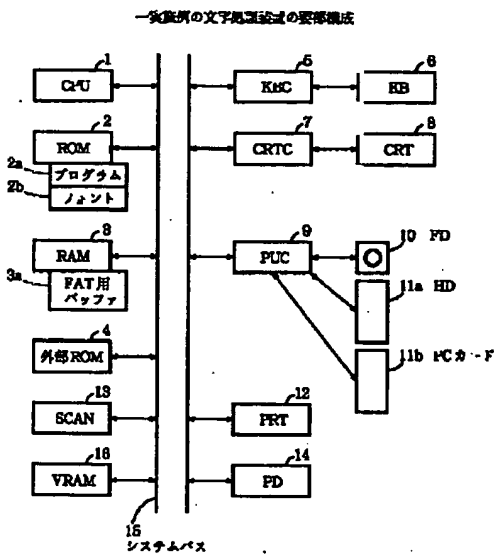
【図6】 一実施例の書き込み処理動作を示すフローチャート

ャート

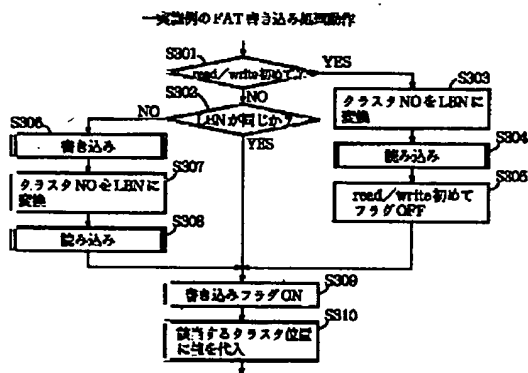
【符号の説明】

- 1 CPU
- 2 ROM
- 3 RAM
- 3a FAT用バッファ
- 9 PUC
- 10 フロッピーディスク (FD)
- 11a ハードディスク (HD)
- 11b PCカード
- 15 システムバス

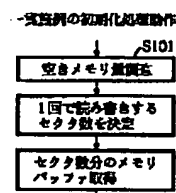
【図1】



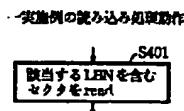
【図4】



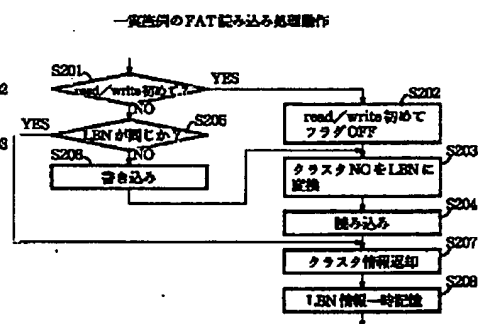
【図2】



【図5】



【図3】



【図6】

